

REMARKS

Claim 5 has been canceled. Claims 1, 6, 9, 20, 23, 33 and 34 have been amended. Claims 1-4, 6-11, 13-31 and 33-38 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Telephone Interview:

During a telephone interview on August 16, 2006 between Examiner Tsai and Applicants' undersigned attorney, Examiner Tsai agreed that the above amendments to the independent claims would overcome the current rejection. Therefore, Applicants assert that the application is in condition for allowance.

Section 102(b) Rejection:

The Examiner rejected claims 1-3, 5, 6, 9, 10, 13-23, 26, 28-31 and 33-38 under 35 U.S.C. § 102(b) as being anticipated by Crater et al. (U.S. Patent 5,146,588) (hereinafter "Crater"). Although Applicants continue to traverse this rejection, Applicants have amended independent claims 1, 20, 33 and 34 to further clarify their distinctive features and submits that these claims as amended are clearly distinguishable over the cited references for at least the following reasons.

As noted in previous responses, Crater provides a redundancy accumulator 301 for accumulating multiple different physical tracks, records or sectors of data stored among storage devices in order to implement a redundancy scheme for recovering lost data (FIG. 4, col. 7, lines 35-60). As disclosed by Crater in col. 8, line 55 – col. 10, line 42, the operation of redundancy accumulator 301 is as follows.

Redundancy accumulator 301 has two modes of operation: read and accumulate. With reference to FIG. 4 and the aforementioned sections of Crater's specification, during the read mode, an address of a byte to be read out of redundancy accumulator 301

is supplied both to redundancy accumulator 301 and pointer memory 302. This causes data to be read from redundancy accumulator 301 via the “output data bus” and additionally causes a logic ‘0’ value to be selected via multiplexer 308 and written into the corresponding byte storage location of pointer memory 302 via the “pointer write data” signal, indicating that the storage location is “stale” and should not be used for a further redundancy calculation. Crater suggests (col. 8, lines 38-43) that redundancy accumulator 301 may be initialized at power on reset time by performing a read access to each byte address, thereby causing each corresponding status bit in pointer memory 302 to be cleared.

During the accumulate mode of operation, Crater discloses that a byte of a physical track of data is received via the “data input bus” and latch 303 while a corresponding address is supplied to redundancy accumulator 301 and pointer memory 302 via the “address bus.” The “read/accumulate” mode signal causes a logic ‘1’ to be written to the addressed location of pointer memory 302 via mux 308, indicating that a valid intermediate or final redundancy calculation result is stored within redundancy accumulator 301. The previous value of pointer memory 302 is read via latch 307 and controls what is written to the addressed location of redundancy accumulator 301. If the pointer read data is a logic ‘0’, indicating that the existing contents of the addressed location of redundancy accumulator 301 are stale, data selector 304 selects only the received byte of data to be written into redundancy accumulator 301. If the pointer read data is a logic ‘1’, indicating that a valid intermediate redundancy calculation result is stored within the addressed location, data selector 304 selects the output of redundancy generator 305, which combines the received byte of data with the previous contents of the addressed location of redundancy accumulator 301 (received via latch 306).

Thus, during the accumulate mode of operation, a series of addresses are applied to redundancy accumulator 301. The corresponding locations are then unconditionally overwritten with either the data received via the data input bus by itself, or the output of a redundancy calculation performed on the received data and the existing contents of redundancy accumulator 301. This process may continue for as many physical tracks of

data as are needed to complete the redundancy calculation. When the calculation is completed, the result may be read in the read mode of operation, which also serves to clear the status of the addressed locations of redundancy accumulator 301 within pointer memory 302 so that the addressed locations may be reused for a subsequent redundancy accumulation operation.

Applicants note that in Crater, when redundancy accumulator 301 is accessed during either read or accumulate mode, there is no concept of a resulting “hit” or “miss.” The contents of the addressed location of redundancy accumulator 301 are unconditionally read and either used in the accumulation operation or output via the output data bus. Crater assumes that whatever value exists within redundancy accumulator 301 is necessarily valid for its intended use. Thus, redundancy accumulator 301 does not function as a cache capable of determining whether needed data is present and fetching the needed data if it is not present. Rather, redundancy accumulator 301 simply functions as a passive buffer configured to store intermediate redundancy results during redundancy operations.

More specifically, Crater neither teaches nor suggests a cache accumulator memory as recited in Applicants’ amended claim 1, which is configured to determine whether a first block operand is stored among a plurality of block storage locations of a cache accumulator memory; in response to determining that the first block operand is stored among the plurality of block storage locations of the cache accumulator memory, provide the first block operand to the functional unit from a first set of block storage locations of the cache accumulator memory and subsequently store the block result generated by the functional unit into the first set of block storage locations; and in response to determining that the first block operand is not stored among the plurality of block storage locations of the cache accumulator memory, initiate a fetch operation to fetch the first block operand from the memory, and upon receiving the first block operand from the memory, store the first block operand within the first set of block storage locations of the cache accumulator memory. As noted above, Crater performs no determination whatsoever of the presence or absence of operands within redundancy

accumulator 301, and does not disclose any aspect of fetching an absent operand from a memory that is cached by the accumulator in response to such a determination. Applicants note that these features are also not taught or suggested by Crater in combination with the other cited references.

Applicants note that similar arguments apply to amended independent claims 20, 33 and 34, which recite limitations similar to amended claim 1. Applicants therefore submit that claims 1, 20, 33 and 34 are distinguishable over the cited references, as are those claims depending from these independent claims.

Section 103(a) Rejection:

The Examiner rejected claims 4, 7, 8 24 and 25 under 35 U.S.C. § 103(a) as being unpatentable over Crater in view of Handy (“The Cache Memory Book”) and claims 11 and 27 as being unpatentable over Crater et al. (U.S. Patent 4,888,679) (hereinafter “Crater”). Applicants traverse this rejection and submit that each of the dependent claims is distinguishable for at least the reasons given above for the independent claims.

CONCLUSION

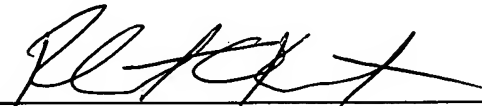
Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-05300/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Other:

Respectfully submitted,


Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: August 21, 2006